On Black-Box Reductions between Predicate Encryption Schemes

Vipul Goyal[1], Virendra Kumar[2,⋆], Satya Lokam[1], and Mohammad Mahmoody[3]

[1] Microsoft Research, Bangalore, India, {vipul, satya}@microsoft.com
[2] Georgia Institute of Technology, Atlanta, GA, USA, virendra@cc.gatech.edu
[3] Cornell University, Ithaca, NY, USA, mohammad@cs.cornell.edu

Abstract. We prove that there is no black-box construction of a threshold predicate encryption system from identity-based encryption. Our result signifies nontrivial progress in a line of research suggested by Boneh, Sahai and Waters (TCC '11), where they proposed a study of the relative power of predicate encryption for different functionalities. We rely on and extend the techniques of Boneh et al. (FOCS '08), where they give a black-box separation of identity-based encryption from trapdoor permutations.

In contrast to previous results where only trapdoor permutations were used, our starting point is a more powerful primitive, namely identity-based encryption, which allows planting exponentially many trapdoors in the public-key by only planting a single master public-key of an identity-based encryption system. This makes the combinatorial aspect of our black-box separation result much more challenging. Our work gives the first impossibility result on black-box constructions of any cryptographic primitive from identity-based encryption.

We also study the more general question of constructing predicate encryption for a complexity class $\mathbb{F}$, given predicate encryption for a (potentially less powerful) complexity class $\mathbb{G}$. Toward that end, we rule out certain natural black-box constructions of predicate encryption for $\mathbf{NC}^1$ from predicate encryption for $\mathbf{AC}^0$ assuming a widely believed conjecture in communication complexity.

Keywords: Predicate Encryption, Black-Box Reductions, Identity-based Encryption, Communication Complexity.

1 Introduction

An encryption scheme enables a user to securely share data with other users. Traditional methods based on Secret-Key Cryptography and Public-Key Cryptography consider the scenarios where a user securely shares data with another *fixed* user whose identity (characterized by the possession of *the* decryption-key) it knows in advance. In particular, in these schemes, there is a bijection between the encryption-key and the decryption-key, fixed by the chosen encryption scheme.

[⋆] Part of the work was done while visiting Microsoft Research, India.

As systems and networks grow in complexity, and in particular with the emergence of the cloud computing, the above viewpoint may be too narrow to cover many important applications. Often, a user might want to encrypt data to be shared with a large set of other users based on some common "property", or attribute, they satisfy. Membership in this set may not be known to the encryptor, or may not even be decidable in advance. Furthermore, a user might want to share data selectively so different users are able to decrypt different parts of that data. To cater to these scenarios, the notion of Predicate Encryption (or Attribute-based Encryption) has recently emerged. Predicate encryption was introduced by Sahai and Waters [31], and further developed in the work of Goyal et al. [17]. It has been the subject of several recent works, e.g., [11, 19, 24, 28, 10]. Predicate encryption is useful in a wide variety of applications; in particular, for fine-grained access control. It has also been a useful technical tool in solving seemingly unrelated problems, e.g., key escrow[15] and user revocation [5] in Identity-based Encryption (IBE). IBE [32, 8, 12] can be seen as the most basic form of a predicate encryption, where the predicate corresponds to a point function.

A predicate encryption scheme is defined in terms of a family $\mathbb{F}$ of Boolean functions (predicates) on a universe $\mathbb{A}$ of attributes. Decryption-keys are associated to a predicate $f \in \mathbb{F}$ and ciphertexts are labeled with (or are created based on) an attribute string $a \in \mathbb{A}$. A user with a decryption-key corresponding to f can decrypt a ciphertext labeled with x if and only if $f(x) = 1$. As argued by Boneh et al. [10], the key challenge in the study of predicate encryption (or Functional Encryption in general) is understanding what classes of functionalities $\mathbb{F}$ can be supported. If we could support any polynomial time computable predicate f, then any polynomial-time access control program that acts over a user's credentials could be supported [10].

Unfortunately, the current state of the art is far from being able to support an arbitrary polynomial-time f. Given this, an important direction Boneh et al. [10] suggested was to understand the relative strengths of predicate encryption schemes with respect to the functionalities they can support: When does a scheme for one functionality imply a scheme for another? In the absence of such a reduction, can we prove that predicate encryption for one functionality is inherently harder than for another? A meaningful approach to address this latter question is via *black-box separations* [18]; see [30, 27] for a comprehensive survey on the topic. A proof that a cryptographic primitive P_1 cannot be constructed given black-box access to another primitive P_2 (and of course without incurring any additional assumptions) can be viewed as an indication that P_1 is in some sense a stronger primitive than P_2. Hence, to construct P_1 one may have to look for more powerful techniques, or stronger assumptions than for P_2 (or try non-black-box reductions). Thus, studying these questions would help us better understand the extent to which techniques for current predicate encryption systems might or might not be useful in obtaining systems for more general functionalities. The broad goal of this work is to make progress toward answering these questions.

Since a predicate encryption scheme has an associated family $\mathbb{F}$ of Boolean functions, a natural way to classify such schemes is according to the complexity class the corresponding family comes from. For example, we can call a scheme $(\mathbb{A}, \mathbb{F})$ an $\mathbf{AC}^0$-PE scheme, if every member of $\mathbb{F}$ can be computed by a constant-depth polynomial size circuit (an $\mathbf{AC}^0$ circuit) on an attribute string from $\mathbb{A}$. Hence, a concrete approach to compare predicate encryption schemes is to ask questions of the kind: *Given a predicate encryption scheme for predicates in complexity class $\mathbb{G}$, can we construct a scheme for predicates in a (potentially larger) complexity class $\mathbb{F}$ in a black-box way?* For example, it is well-known that the circuit class $\mathbf{NC}^1$ is strictly larger than $\mathbf{AC}^0$. Thus a concrete question is: *Is $\mathbf{NC}^1$-predicate encryption provably harder than $\mathbf{AC}^0$-predicate encryption with respect to black-box reductions?* A second aspect of our work is to try to relate (perhaps conjectured) separations among Boolean function complexity classes to black-box separations among the corresponding predicate encryption schemes.

1.1 Our Results

Our main result is a black-box separation of threshold predicate encryption (TPE) from identity-based encryption (IBE) schemes. To our knowledge, this is the first result on the *impossibility* of constructing a cryptographic primitive from IBE in a blackbox manner. Recall that IBE can be viewed as the most basic form of predicate encryption in which the decryption tests exact equality (in other words, the predicate is a point function). Hence, the first natural step in the study of the above question is whether IBE can be used to construct more general predicate encryption systems. Our results show that IBE cannot be used to construct even a basic system for threshold predicates (introduced by Sahai and Waters [31]). We believe that the question of IBE vs. more advanced predicate encryption systems is of special interest. IBE as a primitive is very well studied [8, 12, 7, 6, 34, 14], and constructions of IBE are now known based on a variety of hardness assumptions.

Returning to our more general question, we rule out certain "natural" black-box constructions of predicate encryption for the class $\mathbf{NC}^1$ from predicate encryption for the class $\mathbf{AC}^0$, *assuming* a widely believed conjecture in the area of two-party communication complexity. Given black-box access to a predicate encryption scheme for $(\mathbb{B}, \mathbb{G})$, a natural way to construct a predicate encryption scheme for a "larger" system $(\mathbb{A}, \mathbb{F})$ is to use a a *Sharing-Based Construction* as follows. The decryption-key for an $f \in \mathbb{F}$ is simply the set of decryption keys for a set $S(f) = \{g_1, \dots, g_q\}$ of predicates $g_i \in \mathbb{G}$ from the smaller system. Similarly, for each attribute $a \in \mathbb{A}$, we associate a set $S(a) = \{\alpha_1, \dots, \alpha_q\}$ of attributes from $\mathbb{B}$. To encrypt a message m under an attribute a for the big system, we generate q *shares* $m_1, \dots, m_q$ of m and encrypt m_j under the attribute α_j of the small system. The concatenation of these encrypted shares is the ciphertext of m under a. To decrypt, we try to decrypt each m_j using the decryption keys of each $g_i \in S(f)$. The sharing construction ensures that the shares m_j that are successfully decrypted, if any, in this process suffice to recover m. Thus the sharing-based construction is a rather natural and obvious way to build predicate encryption schemes for more complex functionalities from simpler ones. Our result shows that such a sharing-based construction is impossible if $\mathbb{F}$ is a family of functions in $\mathbf{NC}^1$ and $\mathbb{G}$ is any family of functions from $\mathbf{AC}^0$, assuming certain conjectures in communication complexity. It is worth noting that combinatorial arguments about sharing-based constructions form a core component of our main result on (unrestricted) black-box separation of TPE from IBE.

1.2 Techniques

We build upon and extend the techniques of Boneh et al. [9] (and a follow-up work by Katz and Yerukhimovich [20]) which rule out black-box construction of IBE from Trapdoor Permutations (TDP). Along the way, we also simplify several aspects of their proof. Given a black-box construction of TPE from IBE, our proof proceeds by designing an attack on TPE which succeeds with high probability (in fact arbitrarily close to the completeness probability of the purported TPE scheme). Somewhat more formally, we build an oracle $\mathcal{O}$ relative to which a CCA secure IBE exists, but any purported construction of a TPE relative to this oracle is insecure.

Our analysis of the attack roughly consists of a combinatorial part and a cryptographic part. The combinatorial aspect of our analysis is new and completely different from that in [9]. While the cryptographic part is similar in structure to that of [9], we do make several crucial modifications that makes our attack simpler and analysis cleaner.

A Comparison of the Combinatorial Aspects. At the heart of the proof of [9] is a combinatorial argument as follows. An IBE system obtained by a black-box construction from a TDP must embed in its public parameters the public keys of some permutations of the TDP oracle. The adversary's main goal is to collect all the trapdoors corresponding to these permutations. Such trapdoors are embedded in the decryption keys corresponding to identities in the IBE system. The main point is that there are only $q = \text{poly}(\kappa)$ many permutations planted in the public parameters of the IBE, but they must also encode an exponential number of identities. Therefore, if we look at a sufficiently large set of random identities and their secret keys, and encrypt and decrypt a random message under these identities, during at most q of these decryptions we might encounter a "new" trapdoor (which is planted in the public-key to be used during encryption, but was not discovered during other decryptions). It follows, if we choose our identity set S to be of size $k \cdot q$ (and encrypt and decrypt random messages under them), and then choose an identity $id \xleftarrow{\$} S$ at random from those $q \cdot k$ identities, then with probability at least $1 - 1/k$ there is no new (undiscovered) trapdoor left for this identity id. Therefore, whatever is learned during the decryptions of the encryptions of random messages under the identities $S \setminus \{id\}$, is sufficient to decrypt a message encrypted under id without knowing its decryption-key.

This combinatorial argument immediately suggest the following attack. Get decryption-keys for all but a random identity id_* chosen from a large enough random set $S = id_1, \ldots, id_{k \cdot q}$ of identities. Collect the trapdoors learned from the encryptions of random messages under the identities in $S \setminus id_*$, and their decryptions using the corresponding decryption-keys. Try to decrypt the challenge ciphertext C encrypted under the identity id_*.

In our case, we have a related but more difficult question: what if we start with a more powerful primitive like an IBE and want to construct another "target" predicate encryption scheme? Now the intuition behind the combinatorial argument of [9] completely breaks down. The reason is that in our new setting, by planting only one (master) public-key of the IBE scheme in the public-key of the target predicate encryption, the encryption algorithm potentially has access to an *exponential* number of permutations (each indexed by an identity) whose trapdoors can be planted in the decryption-keys. In fact, each decryption-key of the predicate encryption system might have a unique trapdoor (corresponding to a unique identity derived from the description of the predicate). Hence, one can't hope to learn all trapdoors and use them to decrypt the challenge ciphertext. Thus, roughly speaking, by moving from a trapdoor permutation oracle to various forms of PE oracles such as IBE (as the primitive used in the construction), we are allowing the "universe" of trapdoor permutations planted in the public-key and decryption-keys to be exponentially large (rather than some fixed polynomial). The latter difference is the main reason behind the complications in the combinatorial aspect of our problem, because suddenly the regime of positive results becomes much richer, making the job of proving an impossibility result much more challenging.

Our proof relies on the collusion-resistance property of the predicate encryption. The "hope" that an attack exists comes from the following observations:

- The decryption key for each predicate may still consist of only a polynomial number of IBE decryption-keys.
- Each ciphertext is encrypted using a polynomially large set of identities such that a decryption-key for at least one of these identities is required to decrypt the ciphertext. On the other hand, each ciphertext can be decrypted by keys for an exponential number of different predicates (this follows from the property of a threshold encryption scheme). Call such predicates "related".
- This exponentially large set of related predicates must share an IBE decryption-key since they can decrypt a common ciphertext.

Our attack works by requesting sufficient number of decryption-keys for related predicates (which would still be unable to decrypt challenge ciphertext). Since related predicates share IBE decryption-keys, the adversary is able to collect all "useful" IBE decryption-keys. It is not surprising that the above combinatorial arguments sound as though they could already be used to attack sharing based constructions. Indeed, our core combinatorial lemma (Lemma 10) is used to refute any sharing-based construction of a TPE from an IBE (Corollary 11).

A Comparison of the Cryptographic Aspects. As in [9], turning the combinatorial analysis into a full-fledged impossibility result requires non-trivial black-box separation machinery. For this reason, even though the combinatorial argument of [9] is relatively simple, the full proof is quite complicated. The explanation for the complexity of such proofs is that one has to handle *all possible* constructions using a trapdoor permutation oracle (and not just where, for example, a decryption-key simply consists of decryption keys for various identities).

Although the overall structure of our proof is similar to that of [9], there are several differences in the detailed arguments. In fact, we make some crucial modifications which lead to a more direct attack and cleaner analysis. The first major modification is that our attacker "directly" learns the heavy queries (following the paradigm of [2, 3]). In [9], the attack proceeds by having steps (such as several encryptions of a random bit under the challenge identity, repeating a few steps several times) whose indirect purpose is to learn the heavy queries. Secondly, since we start with an oracle which roughly provides four functionalities (as opposed to the three functionalities of a trapdoor permutation oracle), we need to modify and adapt the techniques of [9] to the new setting. Apart from these, there are significant differences in the manner we compare the various experiments which we believe makes the analysis cleaner and more general. The details regarding these can be found in Section 5 and in the full version [16] where we have deferred most of the proofs due to space constraints.

2 Preliminaries

Notation. For any probabilistic algorithm A, by $y \leftarrow A(x)$ we denote the process of executing A over the input x while using fresh randomness (which we do not represent explicitly) and getting the output y. By a *partial oracle* we refer to an oracle which is defined only for some of the queries it might be asked. By $[x \mapsto y] \in \mathcal{P}$ we mean that $\mathcal{P}(x) = y$ is defined. For a query x and a partial oracle $\mathcal{P}$, we misuse the notation and denote $x \in \mathcal{P}$ whenever an answer for x is defined in $\mathcal{P}$. By $\mathrm{Supp}(X)$ we refer to the support set of the random variable X. For a random variable S whose values are sets, we call an element ϵ-heavy, if $\Pr[x \in S] \geq \epsilon$. The view of any probabilistic oracle algorithm A, denoted as $\mathsf{View}(A)$ refers to its input, private randomness, and oracle answers (which all together determine the whole execution of A).

Definition 1 (Predicate Encryption). *A predicate encryption scheme* $\mathbf{PE}$ *for the predicate set* $\mathbb{F}_\kappa$ *and attribute set* $\mathbb{A}_\kappa$ *with completeness* ρ *consists of four probabilistic polynomial time algorithms* $\mathbf{PE} = (\mathbf{G}, \mathbf{K}, \mathbf{E}, \mathbf{D})$ *such that for every predicate* $f \in \mathbb{F}$*, every attribute* $a \in \mathbb{A}$ *such that* $f(a) = 1$*, and every message* M*, if we do the following steps, then with probability at least* ρ *it holds that* $M' = M$*:* (i). *generate a public-key and a master secret-key* $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathbf{G}(1^\kappa)$*,* (ii). *get a decryption-key* $\mathsf{DK}_f \leftarrow \mathbf{K}(\mathsf{SK}, f)$ *for the predicate* $f \in \mathbb{F}$*,* (iii). *encrypt the message* M *under the attribute* $a \in \mathbb{A}$ *and get* $C \leftarrow \mathbf{E}(\mathsf{PK}, a, M)$*, and finally,* (iv). *decrypt* C *using the decryption-key* DK_f *and get* $M' \leftarrow \mathbf{D}(\mathsf{PK}, \mathsf{DK}_f, C)$*.*

Definition 2 (Neighbor Sets of Predicates and Attributes). *For every set of predicates* $\mathbb{F}$ *and* $f \in \mathbb{F}$*, and for every set of attributes* $\mathbb{A}$ *and* $a \in \mathbb{A}$ *we define the following terminology:*

- $N(f) = \{a \mid a \in \mathbb{A}, f(a) = 1\}$ *and similarly* $\mathbb{N}(a) = \{f \mid f \in \mathbb{F}, f(a) = 1\}$.
- $\deg(f) = |N(f)|$ *and* $\deg(a) = |N(a)|$.

Since we always work with families of algorithms and sets indexed by a security parameter κ, when it is clear from the context we might omit the index κ.

Definition 3 (Security of Predicate Encryption). *Let* $\mathbf{PE} = (\mathbf{G}, \mathbf{K}, \mathbf{E}, \mathbf{D})$ *be a predicate encryption scheme with the predicate set* $\mathbb{F}$ *and the attribute set* $\mathbb{A}$*.* $\mathbf{PE}$ *is said to be CPA secure if for any* $\mathbf{PPT}$ *adversary* $\mathbf{Adv}$ *participating in the experiment below, the probability of* $\mathbf{Adv}$ *correctly outputting the bit b is at most* $1/2 + \text{neg}(\kappa)$*:*

1. **Setup:** *Generate the keys* $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathbf{G}(1^\kappa)$ *and give* PK *to* $\mathbf{Adv}$*.*
2. **Query Keys:** $\mathbf{Adv}$ *adaptively queries some predicates* $f_i \in \mathbb{F}$ *for* $i = 1, 2, \ldots$ *and is given the corresponding decryption-keys* $\mathsf{DK}_i \leftarrow \mathbf{K}(\mathsf{SK}, f_i)$*.*
3. **Challenge:** $\mathbf{Adv}$ *submits an attribute* $a \in \mathbb{A}$ *and a pair of messages* $M_0 \neq M_1$ *of the same length* $|M_0| = |M_1|$ *conditioned on*

$$f_i(a) = 0 \text{ for every predicate } f_i \text{ whose key } \mathsf{DK}_i \text{ is acquired by } \mathbf{Adv} \quad (1)$$

   and is given $C \leftarrow \mathbf{E}(\mathsf{PK}, a, M_b)$ *for a randomly selected* $b \overset{\$}{\leftarrow} \{0, 1\}$*.*
4. $\mathbf{Adv}$ *continues to query keys for predicates subject to condition (1) and finally outputs a bit.*

$\mathbf{PE}$ *is said to be CCA secure if for any* $\mathbf{PPT}$ *adversary* $\mathbf{Adv}$ *participating in a modified experiment (explained next), the probability of* $\mathbf{Adv}$ *correctly outputting the bit b is at most* $1/2 + \text{neg}(\kappa)$*. The modified experiment proceeds identically as the above experiment, except that after Step 3,* $\mathbf{Adv}$ *is also allowed to adaptively query ciphertexts* C_i *for* $i = 1, 2, \ldots$ *encrypted under the attribute a, with the condition that* $C_i \neq C$ *for any i, and he is given the decrypted message* $M \leftarrow \mathbf{D}(\mathsf{DK}_f, C_i)$*, where* $\mathsf{DK}_f \leftarrow \mathbf{K}(\mathsf{SK}, f)$ *is a decryption-key for a predicate f such that* $f(a) = 1$*.*

Definition 4 (Identity-based Encryption [32]). *An Identity Based Encryption scheme is a predicate encryption scheme where* **(1)** *the predicate and attribute sets are equal* $\mathbb{A} = \mathbb{F} = \{0, 1\}^\kappa$ *(and are called the set of identities), and* **(2)** *for every predicate* $f \in \{0, 1\}^\kappa$ *and every attribute* $a \in \{0, 1\}^\kappa$ *we have that* $f(a) = 1$ *if and only if* $f = a$*.*

Definition 5 (Threshold Predicate Encryption [31]). *A Threshold Predicate Encryption with threshold* $0 < \tau < 1$ *(or simply a* τ-$\mathbf{TPE}$*) is a predicate encryption where both the predicate and the attribute sets are equal to* $\{0, 1\}^\kappa$ *and for any predicate* $f \in \{0, 1\}^\kappa$ *and any attribute* $a \in \{0, 1\}^\kappa$ *we have that* $f(a) = 1$ *if and only if* $\langle f, a \rangle \geq \tau \cdot \kappa$ *where* $\langle f, a \rangle$ *is the inner product of the Boolean vectors* $f = (f_1, \ldots, f_\kappa), a = (a_1, \ldots, a_\kappa)$ *defined as* $\langle f, a \rangle = \sum_{i \in [\kappa]} a_i \cdot f_i$*.*

The notion of threshold predicate encryption was defined in [31] and is also known as the *fuzzy* IBE.

3 Sharing-based Constructions and Impossibility Results

In this section, we describe two intuitive and simple approaches to build a predicated encryption scheme using another predicate encryption scheme as a black-box. It is interesting that the simpler of the two, the OR-based approach turns out to be as powerful as the seemingly more general Sharing-based approach. Even though ruling out constructions using these approaches is a weaker impossibility result than an unrestricted black-box separation (as we will do in Section 5), it seems instructive to refute these natural and general approaches to black-box reductions among predicate encryption schemes. In fact, our proof refuting OR-based constructions of TPE from this section forms the combinatorial core of our subsequent proof of a general black-box separation in Section 5. Moreover, the basic approach to building the attack needed in our proof (as well as that in [9]) of the general black-box separation results seems to benefit by keeping the sharing-based constructions in mind. In Section 4, we investigate a new approach to refute sharing-based constructions using (proved or conjectured) separation results in two-party communication complexity. In particular, we can use conjectures in communication complexity to give evidence that $\mathbf{NC}^1$-predicate encryption is strictly harder than $\mathbf{AC}^0$-predicate encryption.

Definition 6. *Let $(\mathbb{F}, \mathbb{A})$ and $(\mathbb{G}, \mathbb{B})$ be two pairs of predicate and attribute sets. We call $S(\cdot)$ a q-*set system *for $(\mathbb{F}, \mathbb{A})$ using $(\mathbb{G}, \mathbb{B})$ if S is a mapping defined over $\mathbb{F} \cup \mathbb{A}$ such that: (1) For every $f \in \mathbb{F}$ it holds that $S(f) \subset \mathbb{G}$, and for every $a \in \mathbb{A}$ it holds that $S(a) \subset \mathbb{B}$, and (2) For every $x \in \mathbb{F} \cup \mathbb{A}$ it holds that $|S(x)| \leq q$.*

Definition 7 (OR-based Construction). *We say there is an OR-based construction with set-size q for the pair of predicate and attribute sets ($\mathbb{F} = \{f_1, \ldots\}$, $\mathbb{A} = \{a_1, \ldots\}$) using another pair ($\mathbb{G} = \{\varphi_1, \ldots\}, \mathbb{B} = \{\alpha_1, \ldots\}$) if there exists a q-set system $S(\cdot)$ for $(\mathbb{F}, \mathbb{A})$ using $(\mathbb{G}, \mathbb{B})$ such that: For every $f \in \mathbb{F}$ and $a \in \mathbb{A}$, if $S(f) = \{\varphi_1, \ldots, \varphi_{d_f}\}$ and $S(a) = \{\alpha_1, \ldots, \alpha_{d_a}\}$, then $f(a) = \bigvee_{i \in [d_f], j \in [d_a]} \varphi_i(\alpha_j)$. We call the OR-based construction* efficient *if the mapping $S(\cdot)$ is efficiently computable.*

The encryption under attribute a of an OR-based construction works by encrypting a message M independently under every $\alpha_i \in S(a)$ and concatenating the corresponding ciphertexts. The decryption key for a predicate f is simply the set of keys DK_j for all $j \in [d_f]$, where DK_j is the decryption key for φ_j.

Lemma 8. *Suppose there exists an efficient OR-based construction for $(\mathbb{F}, \mathbb{A})$ using $(\mathbb{G}, \mathbb{B})$. Then a secure predicate encryption scheme $\mathbf{PE}_1 = (\mathbf{G}_1, \mathbf{K}_1, \mathbf{E}_1, \mathbf{D}_1)$ for $(\mathbb{F}, \mathbb{A})$ with completeness ρ can be constructed (in a black-box way) from any secure predicate encryption scheme $\mathbf{PE}_2 = (\mathbf{G}_2, \mathbf{K}_2, \mathbf{E}_2, \mathbf{D}_2)$ for $(\mathbb{G}, \mathbb{B})$ with completeness ρ.*

Clearly, the OR-based construction of Lemma 8 is not the only way that one can imagine to construct an $\mathbb{F}$-$\mathbf{PE}$ from a $\mathbb{G}$-$\mathbf{PE}$. In fact, as noted also by [20] in the context of using trapdoor permutations, there is a possibility of employing a more complicated "sharing-based" approach that generalizes the OR-based construction. The idea is to use a set system $S(\cdot)$ in a similar way to the OR-based construction, but to encrypt the message M differently: instead of encrypting the message M d_a times, first construct some "shares" $M_1, \ldots, M_{d_a}$ of M, and then encrypt each M_i using α_i. To get the completeness and the security, we need the following two properties.

- *Completeness:* For every $f \in \mathbb{F}$ such that $f(a) = 1$, the set of indices $I_S(a, f) = \{j \mid \exists \varphi \in S(f) \text{ such that } \varphi(\alpha_j) = 1\}$ is rich enough that $\{M_i \mid i \in I_S(a, f)\}$ can be used to reconstruct M.
- *Security:* For every choice of $a_*, f_*, f_1, \ldots, f_k$ for $k = \mathrm{poly}(\kappa)$ such that $f_*(a_*) = 1$ and $f_i(a_*) = 0$ for all $i \in [k]$, it holds that $C_S(a_*, f_*) \not\subseteq \bigcup_{j\in[k]} C_S(a_*, f_j)$, where $C_S(a, f) = \{\alpha_i \mid i \in I_S(a, f)\}$. This is because otherwise the adversary can acquire keys for $f_1, \ldots, f_k$ and use the sub-keys planted in them to decrypt enough of the shares of M_i's and reconstruct M which is encrypted under the attribute a_*.

Despite the fact that the sharing-based approach is more general than the OR-based approach, for the case of polynomial sized sets $q = \mathrm{poly}(\kappa)$, we show that the construction of Lemma 8 is indeed as powerful as any sharing-based approach:

Lemma 9. *There is a sharing based construction for the predicate system $\mathbb{F}$ using $\mathbb{G}$ if and only if there exists an OR-based construction.*

Note that by proving Theorem 19, we shall rule out an OR-based (and hence sharing-based) constructions along the way. A special case of the following combinatorial lemma, Corollary 11, shows that no OR-based (nor sharing-based) construction of τ-**TPE** from **IBE** exists for any constant $0 < \tau < 1$. Moreover, not surprisingly, we will use this lemma in our proof of Theorem 19.

Lemma 10. *Let $\mathbb{F} = \mathbb{A} = \{0,1\}^\kappa$ denote the set of attributes and predicates for τ-**TPE** for a constant $0 < \tau < 1$. Also suppose that the following sets of size at most $q = \mathrm{poly}(\kappa)$ are assigned to $\mathbb{F}$, $\mathbb{A}$, and $\mathbb{F} \times \mathbb{A}$: $S(a)$ for $a \in \mathbb{A}$, $S(f)$ for $f \in \mathbb{F}$, and $S(a, f)$ for $(a, f) \in \mathbb{A} \times \mathbb{F}$. Then, there exists a sampling algorithm Samp that, given an input parameter $\epsilon > 1/\mathrm{poly}(\kappa)$, outputs $k + 1 = \mathrm{poly}(\kappa)$ pairs $(f_*, a_*), (f_1, a_1), \ldots, (f_k, a_k)$ such that with probability at least $1 - \epsilon$ over the randomness of Samp the following holds:*

1. *$f_*(a_*) = 1$ and $f_i(a_i) = 1$ for all $i \in [k]$ (this part holds with probability 1),*
2. *$f_i(a_*) = 0$ for all $i \in [k]$,*
3. *$S(a_*) \cap S(f_*) \cap S(a_*, f_*) \subseteq \bigcup_{i\in[k]} S(a_i, f_i)$.*

Moreover, the algorithm Samp chooses its $k + 1$ pairs without *the knowledge of the set system $S(\cdot)$. Therefore we call Samp an* oblivious *sampler against the predicate structure of τ-**TPE**.*

Note that although $\mathbb{F} = \mathbb{A}$, the sets $S(a)$ for $a \in \mathbb{A}$ and $S(f)$ for $f \in \mathbb{F}$ are potentially different even if a and f represent the same string. Intuitively, the set $S(a)$ refers to the set of sub-attributes (or identities in case of using IBE as the black-box primitive) used during an encryption of a random message under the attribute a, the set $S(f)$ refers to the set of decryption-keys planted in the decryption-key of f, and finally $S(a, f)$ refers to the decryption-keys discovered during the decryption of the mentioned random encryption (under the attribute a) using the generated key for f.

Proof. Let $\mathcal{A}$ be the set of vectors in $\{0,1\}^\kappa$ of normalized Hamming weight τ, namely $\mathcal{A} = \{a \mid a = (a_1, \ldots, a_\kappa) \in \{0,1\}^\kappa, \sum_i a_i = \tau \cdot \kappa\}$. Also let $\mathcal{F}$ be the set of vectors in $\{0,1\}^\kappa$ of normalized Hamming weight $\tau' = \tau + \frac{1-\tau}{2}$. Consider a bipartite graph G with nodes $(\mathcal{A}, \mathcal{F})$ and connect $a \in \mathcal{A}$ to $f \in \mathcal{F}$ iff $f(a) = 1$ according to τ-**TPE** (i.e., the indexes of the nonzero components of a is a subset of those of f). We will later use the fact that G is a regular graph (on its $\mathcal{F}$ side). For any vertex x in G let $N(x)$ be the set of neighbors of x in the graph G. The covering-sampler acts as follows: Choose $p = \mathrm{poly}(\kappa)$ and $h = \mathrm{poly}(\kappa)$ to satisfy $q(\frac{1}{p} + \frac{1}{h} + (1 - \frac{1}{h})^p) < \frac{\epsilon}{2}$ (e.g., this can be done by setting $h = \sqrt{p}$ and choosing p large enough). Choose $f_* \xleftarrow{\$} \mathcal{F}$ at random. Choose $a_*, a_1, \ldots, a_p \xleftarrow{\$} N(f_*)$ at random *with possible repetition* from the neighbors of f_*. For each $i \in [p]$, choose p random neighbors $f_{i1}, \ldots, f_{ip} \xleftarrow{\$} N(a_i)$ of a_i (repetition is allowed). Output the $p^2 + 1$ pairs: $(a_*, f_*), (a_i, f_{ij})_{i \in [p], j \in [p]}$.

Now we prove that with probability at least $1 - \epsilon/2 - \mathrm{neg}(\kappa) > 1 - \epsilon$ the output pairs have the properties specified in Lemma 10.

Property (1) holds by construction.

Since $0 < \tau < \tau' < 1$ are constants, using standard probabilistic arguments one can easily show that the probability of f_{ij} being connected to a_* in G (i.e., $f_{ij}(a_*) = 1$) is $\mathrm{neg}(\kappa)$ (given a_*, a_i are random subsets of f_*, a random superset f_{ij} of a_i is exponentially unlikely to pick all the elements of a_*). Thus (2) holds.

The challenging part is to show that (3) holds, i.e., the following: With probability at least $1 - q(\frac{1}{p} + \frac{1}{\sqrt{p}} + (1 - \frac{1}{\sqrt{p}})^p) \geq 1 - \epsilon/2$ it holds that $S(a_*) \cap S(f_*) \cap S(a_*, f_*) \subset \cup_{ij} S(a_i, f_{ij})$. The proof will go through several claims.

In the following let $h = \sqrt{p}$. For an attribute node $a \in \mathcal{A}$ of G, define $H(a)$ to be the set of "heavy" elements that with probability at least $1/h$ are present in $S(a, f)$ for a random neighbor f of a, i.e., $H(a) = \{x \colon \Pr[x \in S(a, f) \mid f \xleftarrow{\$} N(a)] > 1/h\}$. Note that $H(a)$ is not necessarily a subset of $S(a)$.

Claim. Define BE_1 to be the bad event "$S(a_*) \cap S(a_*, f_*) \not\subseteq H(a_*)$." Then, $\Pr[\mathsf{BE}_1] \leq q/h$.

Proof. Since G is regular on its $\mathbb{F}$ side, conditioned on a fixed a_* the distribution of f_* is still uniform over $N(a_*)$. Now fix a_* and fix an element $b \in S(a_*)$. If b is not in $H(a_*)$, then over the random choice of $f_* \xleftarrow{\$} N(a_*)$, it holds that $\Pr[b \in S(a_*, f_*)] \leq 1/h$. The claim follows by a union bound over the q elements in $S(a_*)$. □

Claim. Define BE_2 to be the bad event "there exists a $b \in S(f_*)$ such that $b \in H(a_*)$ but for every $i \in [p]$, $b \notin H(a_i)$, i.e., $S(f_*) \cap H(a_*) \not\subseteq \cup_i H(a_i)$." Then, $\Pr[\mathsf{BE}_2] \leq q/p$.

Proof. It is enough to bound BE_2 by $1/p$ for a fixed $b \in S(f_*)$ and the claim follows by union bound over the elements of $S(f_*)$. But when $b \in S(f_*)$ is fixed, we can pretend that a_* is chosen at random from the sequence $a_0, \ldots, a_p$ *after* they are chosen and are fixed. In that case BE_2 happens if there is only a unique $j \in \{0, \ldots, p\}$ such that $b \in H(a_j)$ and a_* chooses to be a_j. The latter happens with probability at most $1/(p+1) < 1/p$. □

Claim. Define BE_3 to be the bad event "*given neither* BE_1 *nor* BE_2 *happens,* $S(a_*) \cap S(f_*) \cap S(a_*, f_*) \not\subseteq \cup_{i,j} S(a_i, f_{ij})$." Then, $\Pr[\mathsf{BE}_3] \leq q(1 - 1/h)^p$.

Proof. We assume events BE_1 and BE_2 have not happened and perform the analysis. By $\neg\mathsf{BE}_1$, we have $S(a_*) \cap S(a_*, f_*) \subseteq H(a_*)$. Moreover, since $\neg\mathsf{BE}_2$ holds, any element $b \in S(f_*) \cap H(a_*)$ will be in $H(a_i)$ for at least one $i \in [p]$. Therefore for each $j \in [p]$, $\Pr[b \in S(a_i, f_{ij})] \geq 1/h$ holds by the definition of heavy sets, and thus $b \notin \cup_j S(a_i, f_{ij})$ can hold only with probability at most $(1 - 1/h)^p$. By union bound, the probability that there exists a $b \in S(a_*) \cap S(f_*) \cap S(a_*, f_*)$ such that $b \notin \cup_j S(a_i, f_{ij})$ is bounded by $q(1 - 1/h)^p$. □

From Claims 3, 3, and 3, it follows that (3) fails with probability at most $q(\frac{1}{p} + \frac{1}{h} + (1 - \frac{1}{h})^p) < \frac{\epsilon}{2}$. Therefore, the sampled $[a_*, f_*, \{f_{ij}\}_{i \in [p], j \in [p]}]$ will have the desired properties with probability at least $1 - \text{neg}(\kappa) - \epsilon/2$ which finishes the proof of Lemma 10. □

Using Lemma 10, it is almost straightforward to prove the following.

Corollary 11. *For any constant $0 < \tau < 1$, there is no* OR*-based (nor sharing-based) construction of τ-TPE schemes from IBE schemes.*

4 The Communication Complexity Approach

In this section, we show an alternative general approach to refute sharing-based constructions of predicate encryption schemes using separation results in two-party communication complexity. In particular, using conjectured separations in communication complexity, we prove the impossibility of a sharing-based construction of $\mathbf{NC}^1$-PE from $\mathbf{AC}^0$-PE, thus making some progress toward the question of separating PE schemes based on the complexity classes the underlying predicates come from. On the other hand, we are currently able to apply this approach only to sharing-based constructions rather than to general black-box constructions.

Let $(\mathbb{A}, \mathbb{F})$ be a predicate encryption scheme. W.l.o.g. we identify $\mathbb{A}$ with $\{0,1\}^\kappa$ and think of $\mathbb{F}$ as a family of functions $\{f_b : \{0,1\}^\kappa \to \{0,1\}\}_{b \in \{0,1\}^\kappa}$, i.e., we assume for simplicity that $|\mathbb{F}| = 2^\kappa$ and its members are also indexed by $b \in \{0,1\}^\kappa$. We may abuse this notation and refer to b itself as a member of $\mathbb{F}$. We can then talk about the communications complexity of $\mathbb{F}$ when $b \in \mathbb{F}$ is given to Bob and $a \in \mathbb{A}$ to Alice. We can represent this communication complexity problem by the $\{0,1\}$-matrix with rows indexed by $\mathbb{A}$ and columns by $\mathbb{F}$. With a little more abuse of notation, we denote this matrix also by $\mathbb{F} = (f_b(a))_{a,b}$ and refer to the communication complexity of $\mathbb{F}$. Recall that the essential resource in communication complexity is the number of bits Alice and Bob need to communicate to determine $f_b(a)$. Various models such as deterministic, randomized (public or private coins), nondeterministic, etc., communication complexity can be defined naturally. For details on such models, we refer to the classic book by Kushilevitz and Nisan [22], the paper by Babai et al. [1], and the surveys by Lokam [26] and Lee and Shraibman [23].

To connect communication complexity to OR-based constructions using IBE, we use the model of *Merlin-Arthur (MA) games* in communication complexity:

Definition 12 (Merlin-Arthur Protocols [21]). *A matrix $\mathbb{F}$ is said to have an MA-protocol of complexity $\ell + c$ if there exists a c-bit randomized public-coin verification protocol Π between Alice and Bob such that*

- $\mathbb{F}(a, b) = 1 \Rightarrow \exists w \in \{0,1\}^\ell \quad \Pr[\Pi((a, w), (b, w)) = 1] \geq 2/3,$
- $\mathbb{F}(a, b) = 0 \Rightarrow \forall w \in \{0,1\}^\ell \quad \Pr[\Pi((a, w), (b, w)) = 1] \leq 1/3.$

The MA-complexity of $\mathbb{F}$, denoted $\mathsf{MA}(\mathbb{F})$, is the minimum complexity of an MA protocol for the matrix $\mathbb{F}$.

With this definition, the well-known fact (see, for example, [22]) that $\mathsf{EQUALITY}$ has public coin randomized communication complexity of $O(1)$, and our Definition 7 of OR-construction, the following lemma is easy.

Lemma 13. *Suppose there is an OR-based construction of a predicate encryption scheme $(\mathbb{A}, \mathbb{F})$ using an IBE scheme $(\mathbb{B}, \mathbb{G})$. Then $\mathsf{MA}(\mathbb{F}) = O(\log \kappa)$.*

Using a result due to Klauck [21] that $\mathsf{MA}(\mathsf{DISJOINTNESS}) = \Omega(\sqrt{\kappa})$, we can show

Theorem 14. *For some constant $0 < \tau < 1$, e.g., $\tau = 1/3$, there is no OR-based (and hence no sharing-based) construction of a τ-TPE scheme from IBE.*

To derive separations among stronger predicate encryption schemes based on sharing constructions, we need to recall definitions of languages and complexity classes in two-party communication complexity, in particular, $\mathbf{PH}^{\mathrm{cc}}$ and $\mathbf{PSPACE}^{\mathrm{cc}}$.

Complexity classes in two-party communication complexity are defined in terms of languages consisting of pairs of strings (a, b) such that $|a| = |b|$. Denote by $\{0,1\}^{2*}$ the universe $\{(a, b) : a, b \in \{0,1\}^* \text{ and } |a| = |b|\}$. For a language $L \subseteq \{0,1\}^{2*}$, we denote its characteristic function on pairs of strings of length κ by L_κ. The language L_κ is naturally represented as a $2^\kappa \times 2^\kappa$ matrix with $\{0,1\}$ or ± 1 entries.

Definition 15. *Let $l_1(\kappa), \ldots, l_d(\kappa)$ be nonnegative integers such that $l(\kappa) := \sum_{i=1}^{d} l_i(\kappa) \leq (\log \kappa)^c$ for a fixed constant $c \geq 0$. A language $L \subseteq \{0,1\}^{2*}$ is in $\mathbf{\Sigma}_d^{\mathrm{cc}}$ if there exist $l_1(\kappa), \ldots, l_d(\kappa)$ as above and Boolean functions $\varphi, \psi : \{0,1\}^{\kappa + l(\kappa)} \longrightarrow \{0,1\}$ such that $(a, b) \in L_\kappa$ if and only if $\exists u_1 \forall u_2 \ldots Q_d u_d$ $(\varphi(a, u) \diamondsuit \psi(b, u))$, where $|u_i| = l_i(\kappa), u = u_1 \ldots u_d$, Q_d is $\forall$ for d even and is $\exists$ for d odd, and, $\diamondsuit$ stands for $\vee$ if d is even and for $\wedge$ if d is odd.*

- *By allowing a bounded number of alternating quantifiers, we get an analog of the polynomial time hierarchy:* $\mathbf{PH}^{\mathrm{cc}} = \bigcup_{d>0} \mathbf{\Sigma}_d^{\mathrm{cc}}$.
- *By allowing an unbounded, but at most* $\mathrm{polylog}(\kappa)$ *alternating quantifiers, we get an analog of* **PSPACE***:* $\mathbf{PSPACE}^{\mathrm{cc}} = \bigcup_{c>0} \bigcup_{d \leq (\log \kappa)^c} \mathbf{\Sigma}_d^{\mathrm{cc}}$.

The following lemma shows a connection between the communication complexity class $\mathbf{PH}^{\mathrm{cc}}$ and OR-based constructions using $\mathbf{AC}^0$-predicate encryption.

Lemma 16. *Suppose a predicate encryption scheme* $(\mathbb{A}, \mathbb{F})$ *is obtained by an* OR*-based construction using an* $\mathbf{AC}^0$*-predicate encryption scheme. Then the language given by the sequence of matrices* $\{\mathbb{F}\}_\kappa$ *is in* $\mathbf{PH}^{\mathrm{cc}}$.

Proof. By hypothesis, for a given $f_b \in \mathbb{F}$, we have $\mathbf{AC}^0$ circuits $\varphi_{1b}, \ldots, \varphi_{qb}$ and for a given $a \in \mathbb{A}$, we have $\alpha_{1a}, \ldots, \alpha_{qa}$ such that $f_b(a) = \vee_{i,j} \varphi_{ib}(\alpha_{ja})$. Knowing f_b, Bob can compute the circuit $C_b(z) \equiv \bigvee_{ij} \varphi_{iy}(z_j)$, where $z = (z_1, \ldots, z_q)$, $|z_j| = |\alpha_j|$. Knowing a, Alice can compute $\alpha_a = (\alpha_{1a}, \ldots, \alpha_{qa})$ on which C_b needs to be evaluated.We give a protocol with a bounded number of alternations for $\mathbb{F}$. Let the depth of C_b be d (including the top OR-gate). An existential player will have a move for an OR gate in C_b and a universal player will have a move for an AND gate. Their d moves will describe an accepting path in C_b on α_a. For example, assuming AND and OR gates alternate in successive layers, $\exists w_1 \forall w_2 \cdots Q_d w_d \; \gamma(C_b, w_1, \ldots, w_d)(\alpha_a)$ describes a path in C_b – start with the top OR gate and follow the wire w_1 to the AND gate below and then the wire w_2 from this gate and so on – ending in a gate $\gamma := \gamma(...)$ to witness the claim that $f_b(a) = 1$. Since Bob knows C_b, he can verify the correctness of the path $w_1 w_2 \cdots w_k$ in the circuit and the type of the gate γ given by the path. He then sends the labels of the inputs and the type (AND or OR) of the gate to Alice, who responds with $\gamma(\alpha_a)$. Bob can verify that this will ensure $C_b(\alpha_a) = 1$. On the other hand, if $C_b(\alpha_a) = 0$, then it is easy to see that the existential player will not have a winning strategy to pass verification protocol of Alice and Bob on their inputs a and C_b. It follows that $\mathbb{F}$ has a protocol with at most d alternations and hence $\{\mathbb{F}\}_\kappa \in \mathbf{PH}^{\mathrm{cc}}$. □

This lemma enables us to show the impossibility of OR-based constructions of predicate encryption schemes using $\mathbf{AC}^0$-predicate encryption. In particular,

Theorem 17. *Suppose* $\mathbf{PH}^{\mathrm{cc}} \neq \mathbf{PSPACE}^{\mathrm{cc}}$*. Then, there is no* OR*-based construction of an* $\mathbf{NC}^1$*-PE scheme from any* $\mathbf{AC}^0$*-PE scheme. In particular, there is an* $\mathbf{NC}^1$*-function family* $\mathbb{F}$ *(derived from so-called Sipser functions [33]) such that* $(\mathbb{A}, \mathbb{F})$ *does not have an* OR*-based construction from any* $\mathbf{AC}^0$*-PE scheme.*

However, it is a longstanding open question in communication complexity to separate $\mathbf{PSPACE}^{\mathrm{cc}}$ from $\mathbf{PH}^{\mathrm{cc}}$. Currently it is known that such a separation holds if certain Boolean matrices can be shown to have high rigidity, a connection explained in [29, 25].

Corollary 18. *Suppose Hadamard matrices are as highly rigid as demanded in [29, 25]. Then, predicate encryption defined by the parity functions (arising from Inner Product mod 2 matrix) does not have an* OR*-based construction from any* $\mathbf{AC}^0$*-predicate encryption scheme.*

5 Separating TPE from IBE

In this section, we prove that there is no general black-box construction of threshold predicate encryption schemes from identity-based encryption schemes.

Theorem 19. *Let $\kappa \in \mathbb{N}$ be the security parameter. Then, there exists an oracle $\mathcal{O}$ relative to which CCA secure IBE schemes exist, as per Definition 3. However, for any constant $0 < \tau < 1$, there exists a query-efficient (i.e., that makes at most* $\mathrm{poly}(\kappa)$ *queries to $\mathcal{O}$) adversary* **Adv** *that can break even the* CPA security *of any τ-TPE scheme relative to $\mathcal{O}$, again as per Definition 3. Moreover,* **Adv** *can be implemented in* $\mathrm{poly}(\kappa)$*-time if given access to a* **PSPACE** *oracle, and its success probability can be made arbitrarily close to the completeness of the τ-TPE scheme.*

We will first define our random IBE oracle, $\mathcal{O}_{\mathbf{IBE}}$, also denoted by $\mathcal{O}$ for short, (which trivially implies a CCA secure IBE as outlined in Remark 21), and then break any τ-TPE (with a constant τ) relative to this oracle.

Construction 20 (Randomized oracle $\mathcal{O} = (\mathbf{g}, \mathbf{k}, \mathbf{id}, \mathbf{e}, \mathbf{d})$). *By $\mathcal{O}_\lambda$ we refer to the part of $\mathcal{O}$ whose answers are λ bits, and $\mathcal{O}$ is the union of $\mathcal{O}_\lambda$ for all λ.*

- *The* master-key generating oracle $\mathbf{g} : \{0,1\}^\lambda \mapsto \{0,1\}^\lambda$ *is a random permutation that takes as input a secret-key* $\mathsf{sk} \in \{0,1\}^\lambda$*, and returns a public-key* $\mathsf{pk} \in \{0,1\}^\lambda$.
- *The* decryption-key generating oracle $\mathbf{k} : \{0,1\}^{2\lambda} \mapsto \{0,1\}^\lambda$ *takes as input a secret-key* $\mathsf{sk} \in \{0,1\}^\lambda$ *and an identity $\alpha \in \{0,1\}^\lambda$, and returns a decryption-key* $\mathsf{dk}_\alpha \in \{0,1\}^\lambda$*. We require* $\mathbf{k}(\mathsf{sk}, \cdot)$ *to be a random permutation over* $\{0,1\}^\lambda$ *for every* $\mathsf{sk} \in \{0,1\}^\lambda$.
- *The* identity finding oracle $\mathbf{id} : \{0,1\}^{2\lambda} \mapsto \{0,1\}^\lambda$ *takes as input a public-key* $\mathsf{pk} \in \{0,1\}^\lambda$ *and a decryption-key* $\mathsf{dk} \in \{0,1\}^\lambda$*, and returns the unique α such that* $\mathbf{k}(\mathsf{sk}, \alpha) = \mathsf{dk}$*, where* $\mathsf{sk} = \mathbf{g}^{-1}(\mathsf{pk})$.
- *The* encryption oracle $\mathbf{e} : \{0,1\}^{3\lambda} \mapsto \{0,1\}^\lambda$ *takes as input a public-key* $\mathsf{pk} \in \{0,1\}^\lambda$*, an identity $\alpha \in \{0,1\}^\lambda$ and a message $m \in \{0,1\}^\lambda$, and returns a ciphertext $c \in \{0,1\}^\lambda$. We require* $\mathbf{e}(\mathsf{pk}, \alpha, \cdot)$ *to be a random permutation over* $\{0,1\}^\lambda$ *for every* $(\mathsf{pk}, \alpha) \in \{0,1\}^{2\lambda}$.
- *The* decryption oracle $\mathbf{d} : \{0,1\}^{3\lambda} \mapsto \{0,1\}^\lambda$ *takes as input a public-key* $\mathsf{pk} \in \{0,1\}^\lambda$*, a decryption-key* $\mathsf{dk} \in \{0,1\}^\lambda$ *and a ciphertext $c \in \{0,1\}^\lambda$, and returns the unique m such that* $\mathbf{e}(\mathsf{pk}, \alpha, m) = c$*, where* $\alpha = \mathbf{id}(\mathsf{pk}, \mathsf{dk})$.

By an IBE oracle, we refer to an oracle in the support set of $\mathcal{O}$, $\mathrm{Supp}(\mathcal{O})$*, and by a partial IBE oracle we refer to a partial oracle that could be extended to an oracle in* $\mathrm{Supp}(\mathcal{O})$.

Remark 21 (CCA secure IBE relative to $\mathcal{O}$). To encrypt a bit $b \in \{0,1\}$ under identity α and public-key pk, the encryption algorithm extends b to a λ-bit random string: $m = (b, b_1, \ldots, b_{\lambda-1})$, $b_i \xleftarrow{\$} \{0,1\}$ and gets the encryption $c = \mathbf{e}(\mathsf{pk}, \alpha, m)$. To decrypt, we decrypt c and output its first bit. By independently encrypting the bits of a message $m = (m_1, \ldots, m_n)$, with $n = \mathrm{poly}(\kappa)$, and using a standard hybrid argument, one can generalize the scheme to arbitrarily long messages. This construction is only CPA secure, where any adversary has advantage at most $2^{-\Theta(\kappa)}$. But, this can easily be transformed in a black-box manner into a CCA secure construction, without incurring any additional assumptions, using the Fujisaki-Okamoto transform [13] in the random oracle model [4]. We note that even though $\mathcal{O}$ is not exactly a random oracle, for our purposes it suffices to use one of the sub-oracles of $\mathcal{O}$ as a random oracle in the above transform.

Now we present an attack that aims to break any τ-TPE in an $\mathcal{O}$-relativized world by asking only $\mathrm{poly}(\kappa)$ queries to the random IBE oracle $\mathcal{O}$, where κ is the security parameter of the τ-TPE scheme. We prove the query-efficiency and the success probability of our attack in the full version [16]. Similar to the attack of [9], our attack can easily be implemented in $\mathrm{poly}(\kappa)$-time if $\mathbf{P} = \mathbf{PSPACE}$[4], and the relativizing reductions can be ruled out by adding a **PSPACE** oracle to $\mathcal{O}$.

We first note that any black-box construction of τ-TPE schemes from IBE schemes can potentially call the oracle $\mathcal{O}_\lambda$ over different values of λ which are potentially different from the security parameter of the τ-TPE scheme itself. However, similar to [9], we assume that the τ-TPE scheme asks its queries to $\mathcal{O}_\lambda$ only for one value of λ. This assumption is purely to simplify our presentation of the attack and its analysis, and all the arguments below extend to the general case (of asking queries over any parameter $\lambda > \log s$) in a straightforward way.

We also assume that λ is large enough in the sense that $2^\lambda > s$ for an arbitrarily large $s = \mathrm{poly}(\kappa)$ that can be chosen in the description of the attack. The reason for the latter assumption is that the adversary can always ask and learn *all* the oracle queries to $\mathcal{O}$ that are of logarithmic length $O(\lambda) = O(\log \kappa)$, simply because there are at most $2^{O(\lambda)} = \mathrm{poly}(\kappa)$ many queries of this form.[5]

Construction 22 (Adv Attacking the Scheme τ-TPE$^{\mathcal{O}}$). *The parameters are as follows. q: the total number of queries asked by the components of the scheme τ-**TPE** all together, κ: the security parameter of τ-**TPE**, $\epsilon = 1/\mathrm{poly}(\kappa)$ and $s = \mathrm{poly}(\kappa)$: input parameter to the adversary **Adv**, $\lambda \leq \mathrm{poly}(\kappa)$: the parameter which determines the output length of the queries asked by the components of τ-**TPE** to the oracle $\mathcal{O}$. It is assumed that $2^\lambda > s$ for some $s = \mathrm{poly}(\kappa)$ to be chosen later. Our adversary **Adv** executes the following.*

1. **Sampling Predicates and Attributes: Adv** *executes the sampling algorithm* Samp *of Lemma 10 with the parameter ϵ, over the predicate structure of τ-**TPE**, to get $k+1$ pairs $(a_*, f_*), \{(a_i, f_i)\}_{i \in [k]}$. Recall that this sampling is done only by knowing the predicate structure of τ-**TPE** and is independent of the actual implementation of the scheme. It can be done, for example, without the knowledge of* PK.

[4] A good “approximation” of the attack can also be implemented assuming $\mathbf{P} = \mathbf{NP}$.

[5] In [9] a scheme that asks such queries is called “degenerate” and is handled similarly.

2. **Receiving the Keys:** **Adv** *receives from the challenger: the public-key* PK *and the decryption-keys* $\{\mathsf{DK}_i\}_{i\in[k]}$, *where* DK_i *is the generated decryption-key for* f_i. *We also assume that* DK_* *is generated by the challenger, although* **Adv** *does not receive it. Let* V *be the view of the algorithms executed by the challenger so far that generated the keys* $\mathsf{PK}, \mathsf{DK}_*, \mathsf{DK}_1, \ldots, \mathsf{DK}_k$. *Let* $Q(V)$ *be the partial oracle consisting of the queries (and their answers) specified in* V. *By writing in the bold font* $\mathbf{V}$, *we refer to* V *as a random variable.*
3. **Encrypting Random Bits:** *For all* $i \in [k]$, **Adv** *chooses a random bit* $d \overset{\$}{\leftarrow} \{0,1\}$, *computes the encryption* $C_i \leftarrow \mathbf{E}(\mathsf{PK}, a_i, d)$, *and then the decryption* $\mathbf{D}(\mathsf{PK}, \mathsf{DK}_i, C_i)$. *Let* $\mathcal{L}_0$ *be the partial oracle consisting of the oracle queries (and their answers) that* **Adv** *observes in this step.*
4. **Learning Heavy Queries:** *This step consists of some internal rounds. For* $j = 1, 2, \ldots$ *do the following. Let* $\mathcal{L}_j$ *be the partial oracle consisting of the oracle queries (and their answers) that* **Adv** *has learned about* $\mathcal{O}$ *till the end of the* j*'th round*[6] *of this learning step. Let* $\mathbf{V}_j = (\mathbf{V} \mid \mathcal{L}_j, \mathsf{PK}, \{\mathsf{DK}\}_{i\in[k]})$ *be the distribution of the random variable* $\mathbf{V}$ *(also including the randomness of* $\mathcal{O}$*) conditioned on the knowledge of* $(\mathcal{L}_j, \mathsf{PK}, \{\mathsf{DK}\}_{i\in[k]})$. *For a partial oracle* $\mathcal{P}$, *let* $\overline{\mathcal{P}}$ *denote its closure*[7]. *Now, if there is any query* x *such that* $x \notin \mathcal{L}_j$ *but* $\Pr[x \in \overline{Q(\mathbf{V}_j)}] \geq \epsilon$, **Adv** *asks the lexicographically first such* x *from the oracle* $\mathcal{O}$, *sets* $\mathcal{L}_{j+1} = \mathcal{L}_j \cup (x, \mathcal{O}(x))$, *and goes to round* $j+1$. *In other words, as long as there is any new query* x *that is* ϵ*-heavy to be in the closure of the queries of the view of the key-generations,* **Adv** *asks such a query* x. *If no such query exists,* **Adv** *breaks the loop and goes to the next step.*

   (Note that the above and the following steps may require a **PSPACE***-complete oracle to be implemented efficiently.)*
5. **Guessing Challenger's View:** *Let* $\mathcal{L}$ *be the partial oracle consisting of the oracle queries (and their answers) that* **Adv** *learned in Steps 3 and 4 (i.e.,* $\mathcal{L} = \mathcal{L}_\ell$, *where* $\overline{Q(\mathbf{V}_\ell)}$ *had no* ϵ*-heavy queries to be learned). Let* $\mathbf{V_{chal}} = (\mathbf{V} \mid \mathcal{L}, \mathsf{PK}, \{\mathsf{DK}_i\}_{i\in[k]})$, *and sample* $V' \overset{\$}{\leftarrow} \mathbf{V_{chal}}$. *Let* SK' *and* DK'_* *be in order, the "guessed" values for the secret-key and the decryption-key of* f_* *determined by the sampled* V'. *We note that by definition the other keys* $\mathsf{PK}', \{\mathsf{DK}'_i\}_{i\in[k]}$ *determined by* V' *are the same as the ones that* **Adv** *has received:* $\mathsf{PK}, \{\mathsf{DK}_i\}_{i\in[k]}$.
6. **Receiving the Challenge and the Final Decryption:** **Adv** *receives* $C_*(= \mathbf{E}^{\mathcal{O}}(\mathsf{PK}, a_*, b))$ *for a random bit* $b \in \{0,1\}$. *Then,* **Adv** *uses the oracle* $\mathcal{O}'$ *defined below and outputs the decrypted value* $b' \leftarrow \mathbf{D}^{\mathcal{O}'}(\mathsf{PK}, \mathsf{DK}'_*, C_*)$ *as his guess about the bit* b.

   The Oracle $\mathcal{O}'$*: At the beginning of the decryption of Step 6, the partially defined oracle* $\mathcal{O}'$ *is equal to* $\mathcal{L} \cup Q(V')$, *namely the learned queries (and*

[6] Step 3 can be thought of as the 0'th round.

[7] Informally, the closure of a partial oracle is a superset consisting of all the queries (in addition to the partial oracle itself) that are dependent on the queries in the partial oracle (or its closure), e.g. if the partial oracle contains queries $[\mathbf{g}(\mathsf{sk}) = \mathsf{pk}]$ and $[\mathbf{k}(\mathsf{sk}, \alpha) = \mathsf{dk}]$ then its closure must also contain the query $[\mathbf{id}(\mathsf{pk}, \mathsf{dk}) = \alpha]$. Please refer to the full version [16] for a formal definition.

their answers) together with the guessed ones specified in V'. Afterwards, if a new query x is asked: (i) if $x \in \mathcal{O}'$, return $\mathcal{O}'(x)$, otherwise (ii) if $x \in \overline{\mathcal{O}'}$, then return $y = \overline{\mathcal{O}'}(x)$ and add (x, y) to $\mathcal{O}'$, and finally (iii) if $x \notin \overline{\mathcal{O}'}$, ask x from $\mathcal{O}$ and add $(x, \mathcal{O}(x))$ to $\mathcal{O}'$.

This finishes the description of our attack. We prove the query-efficiency and the success probability of our attack in the full version [16].

Acknowledgement. We thank Brent Waters for suggesting to us the problem of separating predicate encryption from IBE and for useful collaborations in initial stages of this work. We thank Yevgeniy Vahlis for clarifying several doubts in [9] and the anonymous reviewers for their valuable comments. Virendra Kumar was supported in part by Alexandra Boldyreva's NSF CAREER award 0545659 and NSF Cyber Trust award 0831184. Mohammad Mahmoody was supported in part by NSF Award CCF-0746990, AFOSR Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2- 0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US government.

References

1. L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory (preliminary version). In *FOCS*, pages 337–347, 1986.
2. B. Barak and M. Mahmoody-Ghidary. Lower Bounds on Signatures From Symmetric Primitives. In *FOCS*, pages 680–688, 2007.
3. B. Barak and M. Mahmoody-Ghidary. Merkle Puzzles Are Optimal - An $O(n^2)$-Query Attack on Any Key Exchange from a Random Oracle. In *CRYPTO*, pages 374–390, 2009.
4. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
5. A. Boldyreva, V. Goyal, and V. Kumar. Identity-based encryption with efficient revocation. In *ACM Conference on Computer and Communications Security*, pages 417–426, 2008.
6. D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.
7. D. Boneh and X. Boyen. Secure Identity Based Encryption Without Random Oracles. In *CRYPTO*, pages 443–459, 2004.
8. D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.
9. D. Boneh, P. A. Papakonstantinou, C. Rackoff, Y. Vahlis, and B. Waters. On the Impossibility of Basing Identity Based Encryption on Trapdoor Permutations. In *FOCS*, pages 283–292, 2008.
10. D. Boneh, A. Sahai, and B. Waters. Functional Encryption: Definitions and Challenges. In *TCC*, pages 253–273, 2011.
11. D. Boneh and B. Waters. Conjunctive, Subset, and Range Queries on Encrypted Data. In *TCC*, pages 535–554, 2007.

12. C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In *IMA Int. Conf.*, pages 360–363, 2001.
13. E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *CRYPTO*, pages 537–554, 1999.
14. C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.
15. V. Goyal. Reducing Trust in the PKG in Identity Based Cryptosystems. In *CRYPTO*, pages 430–447, 2007.
16. V. Goyal, V. Kumar, S. Lokam, and M. Mahmoody. On Black-Box Reductions between Predicate Encryption Schemes. In *TCC*, 2012. Full version available at: http://eprint.iacr.org.
17. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.
18. R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In *STOC*, pages 44–61, 1989.
19. J. Katz, A. Sahai, and B. Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *EUROCRYPT*, pages 146–162, 2008.
20. J. Katz and A. Yerukhimovich. On Black-Box Constructions of Predicate Encryption from Trapdoor Permutations. In *ASIACRYPT*, pages 197–213, 2009.
21. H. Klauck. Rectangle Size Bounds and Threshold Covers in Communication Complexity. In *IEEE Conference on Computational Complexity*, pages 118–134, 2003.
22. E. Kushilevitz and N. Nisan. *Communication complexity.* Cambridge University Press, 1997.
23. T. Lee and A. Shraibman. Lower Bounds in Communication Complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–398, 2009.
24. A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *EUROCRYPT*, pages 62–91, 2010.
25. S. V. Lokam. Spectral Methods for Matrix Rigidity with Applications to Size-Depth Trade-offs and Communication Complexity. *J. Comput. Syst. Sci.*, 63(3):449–473, 2001.
26. S. V. Lokam. Complexity Lower Bounds using Linear Algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1-2):1–155, 2009.
27. M. Mahmoody-Ghidary and A. Wigderson. Black Boxes, Incorporated, 2009. http://www.cs.cornell.edu/ mohammad/files/papers/BlackBoxes.pdf.
28. T. Okamoto and K. Takashima. Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In *CRYPTO*, pages 191–208, 2010.
29. A. Razborov. On Rigid Matrices (in Russian), 1989. http://people.cs.uchicago.edu/ razborov/rigid.pdf.
30. O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of Reducibility between Cryptographic Primitives. In *TCC*, pages 1–20, 2004.
31. A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *EUROCRYPT*, pages 457–473, 2005.
32. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO*, pages 47–53, 1984.
33. M. Sipser. Borel Sets and Circuit Complexity. In *STOC*, pages 61–69, 1983.
34. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.